

REMARKS:

Status

Claims 1 to 29 are pending. The independent claims, namely claims 1, 7, 24 and 27, are amended herein. Reconsideration and further examination are respectfully requested.

Claim Rejections

Claims 1 to 29 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,535,878 (Guedalia).

Claims 1 to 6 and 12 to 17: Claim 1 is reproduced below:

1. A method including  
simulating a plurality of dynamically-allocated threads using a  
single statically-allocated thread; and  
maintaining state information used by each dynamically-allocated  
thread in variables maintained by said statically-allocated thread.

As discussed in previous papers filed in this case, the applied Guedalia reference is not seen to disclose or to suggest the foregoing features of claim 1, at least with respect to “simulating a plurality of dynamically-allocated threads using a single statically-allocated thread.”

The Advisory Action stated the following in response to an earlier presentation of this argument:

Applicant alleges the threads in Guedalia are actually dynamically-allocated threads while in contrast, claim 1 uses “a single statically-allocated thread” to simulate “a plurality of dynamically-allocated

threads.” Examiner respectfully disagrees because Guedalia provides a thread management system, where the requests are processed by a single thread (i.e., static thread) to generate dynamically allocated threads (see col. 9, line 52 to col. 10, line 7).

In response, Applicant completely agrees with the Examiner’s characterization that “Guedalia provides a thread management system, where the requests are processed by a single thread (i.e., static thread) to generate dynamically allocated threads.” However, generating threads is entirely different from simulating threads. In fact, this language actually supports Applicant’s argument: something that generates other actual dynamically-allocated threads has no need to simulate those threads.

Of course, generation and simulation of threads can co-exist in a system and would be within the scope of the invention as claimed. However, Applicant submits that Guedalia simply does not discuss simulation of threads.

Guedalia does discuss that each thread in a multitasking operating system can simulate a separate application (col. 3, lines 63 to 65). Guedalia also discusses a server test that was run using requests from simulated clients (col. 20, lines 44 to 47). However, no other mention is made in Guedalia of simulation, and in particular no mention is made of *simulating threads*.

Furthermore, Guedalia is not seen to disclose or to suggest claim 1’s feature of “maintaining state information used by each dynamically-allocated thread in variables maintained by said statically-allocated thread.”

In this regard, Guedalia does teach “a Manage Threads step 126 [that] is constantly monitoring the active threads in thread pool 124.” Guedalia, col. 22, lines 12 to 20. However, no mention is made of those active threads themselves using any information maintained by the Manage Threads step. Rather, Guedalia teaches using the information to decide how to manage those threads.

In view of the foregoing, reconsideration and withdrawal are respectfully requested of the § 102(e) rejection of claim 1 and its dependent claims. Allowance of these claims also is requested.

Claims 7 to 11 and 19 to 23: Claim 7 is reproduced below:

7. Apparatus including a file server system having a single statically-allocated thread including a plurality of simulated dynamically-allocated threads, said statically-allocated thread maintaining variables that maintain state information used by each said simulated thread.

The applied Guedalia reference is not seen by Applicant to disclose or to suggest the foregoing features of claim 7, at least with respect to “a single statically-allocated thread including a plurality of simulated dynamically-allocated threads” and with respect to “said statically-allocated thread maintaining variables that maintain state information used by each said simulated thread.” Accordingly, reconsideration and withdrawal are respectfully requested of the § 102(e) rejection of claim 7 and its dependent claims. Allowance of these claims also is requested.

Claims 24 to 26: Claim 24 is reproduced below:

24. A method of implementing a plurality of simulated dynamically-allocated threads using a single statically-allocated thread, comprising:

using a scheduler implemented by said single statically-allocated thread to call thread blocks for said plurality of simulated dynamically-allocated threads; and

maintaining state information used by each of said plurality of simulated dynamically-allocated threads in variables maintained by said statically-allocated thread.

The applied Guedalia reference is not seen by Applicant to disclose or to suggest the foregoing features of claim 24, at least with respect to “said plurality of simulated dynamically-allocated threads” and with respect to “maintaining state information used by each of said plurality of simulated dynamically-allocated threads in variables maintained by said statically-allocated thread.” Accordingly, reconsideration and withdrawal are respectfully requested of the § 102(e) rejection of claim 24 and its dependent claims. Allowance of these claims also is requested.

As an ancillary matter, Applicant notes that in the previously filed Response to Final Office Action, Applicant argued that Guedalia “is not seen by Applicant to disclose or to suggest the foregoing features of claim 24, at least with respect to ‘using a scheduler implemented by said single statically-allocated thread to call thread blocks for said plurality of simulated dynamically-allocated threads.’” Applicant wishes to clarify that the argument was not that Guedalia failed to disclose using a scheduler, but rather failed to disclose the particular scheduler recited by claim 24, which “call[s] thread blocks for *said plurality of simulated dynamically-allocated threads*.”

Claims 27 to 29: Claim 27 is reproduced below:

27. Apparatus including a server that implements a plurality of simulated dynamically-allocated threads using a single statically-allocated thread, comprising:

a processor that executes a scheduler implemented by said single statically-allocated thread to call thread blocks for said plurality of simulated dynamically-allocated threads; and

memory that stores state information used by each of said plurality of simulated dynamically-allocated threads in variables maintained by said statically-allocated thread.

The applied Guedalia reference is not seen by Applicant to disclose or to suggest the foregoing features of claim 27, at least with respect to “said plurality of simulated dynamically-allocated threads” and with respect to “memory that stores state information used by each of said plurality of simulated dynamically-allocated threads in variables maintained by said statically-allocated thread.” Accordingly, reconsideration and withdrawal are respectfully requested of the § 102(e) rejection of claim 27 and its dependent claims. Allowance of these claims also is requested.

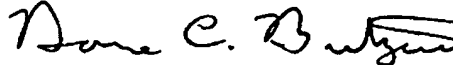
As an ancillary matter, Applicant notes that in the previously filed Response to Final Office Action, Applicant argued that Guedalia “is not seen by Applicant to disclose or to suggest the foregoing features of claim 27, at least with respect to ‘a scheduler implemented by said single statically-allocated thread to call thread blocks for said plurality of simulated dynamically-allocated threads.’” Applicant wishes to clarify that the argument was not that Guedalia failed to disclose using a scheduler, but rather failed to disclose the particular scheduler recited by claim 27, which “call[s] thread blocks for *said plurality of simulated dynamically-allocated threads*.”

Closing

In view of the foregoing amendments and remarks, the entire application is believed to be in condition for allowance, and such action is respectfully requested at the Examiner's earliest convenience.

Applicant's undersigned attorney can be reached at (614) 486-3585. All correspondence should continue to be directed to the address indicated below.

Respectfully submitted,



Dated: March 22, 2004 (Monday)

Dane C. Butzer  
Reg. No. 43,521

The Swernofsky Law Group  
P.O. Box 390013  
Mountain View, CA 94039-0013  
(650) 947-0700